MetaMap: Mapping Text to the UMLS® Metathesaurus®

Alan R. Aronson

March 6, 1996

The task of automatically determining the concepts referred to in text is a common one. It occurs in SPECIALIST™ as a prerequisite to improving the retrieval of relevant MEDLINE® citations based on queries formulated in English. In this case the text consists not only of user queries but also the titles and abstracts of MEDLINE citations; and the concepts to be found in the text are those of the UMLS Metathesaurus. The task becomes one of mapping text to the Metathesaurus (referred to subsequently as Meta). This report describes the development of MetaMap, a program for automatically mapping biomedical text to Meta. Section 1 contains the results of manually examining the mapping problem for a small collection of utterances. It provides the basis for defining a strategy for automatically mapping to Meta as outlined in Section . The automatic approach is characterized by determining how to map from the noun phrases discovered by the SPECIALIST minimal commitment parser to appropriate concepts in Meta. The results of such a mapping can be used to normalize the text so that each referenced concept is represented uniquely. Details of the MetaMap implementation are given in Section 3 through Section 7. It should be noted that the MetaMap algorithms are not specific to the biomedical domain and can be generalized to any domain with adequate knowledge sources.

1. A Preliminary Examination of the Mapping Problem

In order to determine the scope of the problem of mapping text to Meta, a set of 99 utterances (16 queries and 83 citation titles) was taken from the NLM Test Collection. The SPECIALIST minimal commitment parser was applied to the utterances producing 301 noun phrases. Each of the phrases was manually mapped to the 1992 version of Meta and classified into one of four categories based on how well it maps to Meta.[1] Membership in a category is determined by lexical properties of the mapping as defined below; and in each case inflectional and spelling variation are ignored:

- Simple match—the noun phrase maps exactly to a Meta string. For example, *intensive care unit* maps to Intensive Care Units.
- Complex match—there is a partitioning of the words in the noun phrase so that each element of the partition has a Simple match to Meta. For example, *intensive care medicine* maps to the two Meta terms Intensive Care and Medicine
- Partial match—the noun phrase maps to a Meta string in such a way that at least one word of either the noun phrase or the Meta string (or both) does not participate in the mapping. Partial matches have the following variations:

  Normal partial match—The simplest type of partial match occurs when a Meta string maps part of the noun phrase without gaps in what it *does* map. For example, *liquid crystal thermography* maps to Thermography where the mapping does not involve *liquid crystal*. Similarly, *cochlear implant subjects* maps to Cochlear Implant where *subjects* is not involved. Normal partial matches provide good results for the part of the noun phrase involved.

  Gapped partial match—Gapped partial matches involve a gap either in the noun phrase or Meta string or both. For the mapping of *ambulatory monitoring* to AMBULATORY CARDIAC MONITORING, the gap CARDIAC occurs in the Meta string. For the mapping of *obstructive sleep apnea* to Obstructive Apnea, the gap *sleep* occurs in the noun phrase. And for the mapping of *continuous pump-driven hemofiltration* to Continuous Arteriovenous Hemofiltration, gaps occur in both. Gapped partial matches often provide better results than normal partial matches because of their greater matching involvement. However, when the gap occurs in the Meta string, the string tends to be too specific.

  Overmatch—An overmatch occurs when a match does not involve words at either or both ends of the Meta string. An overmatch is similar to a normal partial match except that part of the Meta string is uninvolved in the mapping. For example, the Meta string Postoperative Complications is an overmatch for *ocular complications*. The phrase *application* has many overmatches including Job Application, Heat/Cold Application and Medical Informatics Application. Overmatches almost always give poor results unless browsing is the object of the mapping.
- No match—no part of the noun phrase maps to any string in Meta.[2]

The categories above are listed in order of the strength of the mapping, a simple match being the strongest. It should be emphasized, however, that the semantic or conceptual quality of the mapping varies widely. Even simple matches can map text to unrelated Meta terms. For example, the noun phrase *the numeric values* maps to the Meta concept Values with semantic type Qualitative

1. All Meta examples in this document are taken from 1992 Meta unless otherwise specified.
2. It is becoming very difficult to find such matches. For example, the phrase *improvement* had no mappings to 1992 Meta, but maps to Improved and also to the eleven-word 1994 Meta concept Coreoplasty by photocoagulation, one or more sessions for improvement of vision.

Concept. In Meta, Values is a term from psychology referring to social values; it is not a quantitative concept. Furthermore, even when Meta contains the correct concept, that concept may be ambiguous. For example, Meta contains two *ventilation* concepts, one related to air flow in buildings and the other related to respiration.

Some examples from the manual study illustrating the types of match just discussed are given below. In each case a phrase and one or more Meta terms are listed together with the type of match. Note that each phrase generally maps to more Meta terms than shown.

- *computerized system* —> Computer Systems (simple) and Computerized Medical Record Systems (gapped partial);
- *indications* —> Indicators (simple);
- *ischaemic limbs*—> Ischemic and Limbs (complex);
- *diabetic foot* —> Diabetes and Foot (complex), DIABETIC FOOT CARE (overmatch), and Diabetic Foot (simple);
- *obstructive sleep apnea* —> Obstructive Apnea (gapped partial) and Sleep apnea (normal partial);
- *membrane plasma filtration* —> Membranes, Plasma, and Filtration (complex);
- *inferior vena caval stent filter* —> Inferior Vena Cava Filter and Stent (complex);
- *laboratory tests* —> laboratory Tests (simple) but also TEST[1] (normal partial);
- *cardiokymography* —> Kymography (simple); and
- *phrenic motoneurones* —> Motor Neurons (normal partial).

The results of manually mapping the 301 noun phrases to Meta are summarized in Table 1.[2] [Note that 70 percent of the 113 partial matches involved the head of the noun phrase.]

Lexical Mapping Category	Count	Percent
Simple match	91	30%
Complex match	24	8%
Partial match	113	38%
No match	73	24%
Total	301	100%

Table 1. Summary of Noun Phrase Mappings to Meta

1. In Meta, TEST is the chemical Ethanesulfonic acid, 2-((2-hydroxy-1,1-bis(hydroxymethyl)ethyl)amino)-, mixt. with 2-amino-2-(hydroxymethyl)-1,3-propanediol.

2. The results shown were obtained using the 1992 version of the Metathesaurus; later versions would give better results due to increased domain coverage.

2. The Basic Mapping Strategy

The experience gained from the manual mapping exercise described above led naturally to the following strategy for accomplishing the mapping automatically. Perform the following steps for each textual utterance:

1. Parse the text into noun phrases and perform the remaining steps for each phrase;[1]
2. Generate the variants for the noun phrase where a variant essentially consists of one or more noun phrase words together with all of its spelling variants, abbreviations, acronyms, synonyms, inflectional and derivational variants, and meaningful combinations of these;
3. Form the *candidate set* of all Meta strings containing one of the variants;
4. For each candidate, compute the mapping from the noun phrase and calculate the strength of the mapping using an evaluation function. Order the candidates by mapping strength; and
5. Combine candidates involved with disjoint parts of the noun phrase, recompute the match strength based on the combined candidates, and select those having the highest score to form a set of best Meta mappings for the original noun phrase.

Descriptions of steps 2-5 of the mapping strategy are given in the next four sections.

3. Noun Phrase Variants

The Meta mapping algorithm begins by computing a set of variant generators for each noun phrase discovered by the parser. A variant generator is any *meaningful* subsequence of words in the phrase where a subsequence is meaningful if it is either a single word or occurs in the SPECIALIST lexicon. For example, the variant generators for the noun phrase *of liquid crystal thermography*[2] are *liquid crystal thermography*, *liquid crystal*, *liquid*, *crystal* and *thermography* (prepositions, determiners, conjunctions, auxiliaries, modals, pronouns and punctuation are ignored). Note the multi-word generators. A simpler example which will be used throughout the sequel is based on the noun phrase *ocular complications.*[3] Its generators are simply *ocular* and *complications*.

The approach taken in computing variants is a canonicalization approach. This simply means that a variant represents not only itself but all of its inflectional and spelling variants.[4] Collapsing inflectional and spelling variants results in significant computational savings. Variants are

1. Parsing is accomplished using the SPECIALIST minimal commitment parser which produces a high-level syntactic analysis rather than a full syntactic analysis. The parser optionally uses the Xerox Part-of-speech tagger which assigns syntactic labels to all textual items. The parser is very good at determining the simple noun phrases in text; and the errors it does make are normally inconsequential to MetaMap. The tagger also improves parsing results.

2. A simplified syntactic analysis for *of liquid crystal thermography* is `[prep(of), head(liquid crystal thermography)]`.

3. A simplified syntactic analysis for *ocular complications* is `[mod(ocular), head(complications)]`.

4. A spelling variant of a word is just a variant having the same principal part as the word. For example, *haemorrhaged* is a spelling variant of *hemorrhaged*.

computed for each of the variant generators according to the scheme pictured in Figure 1. The

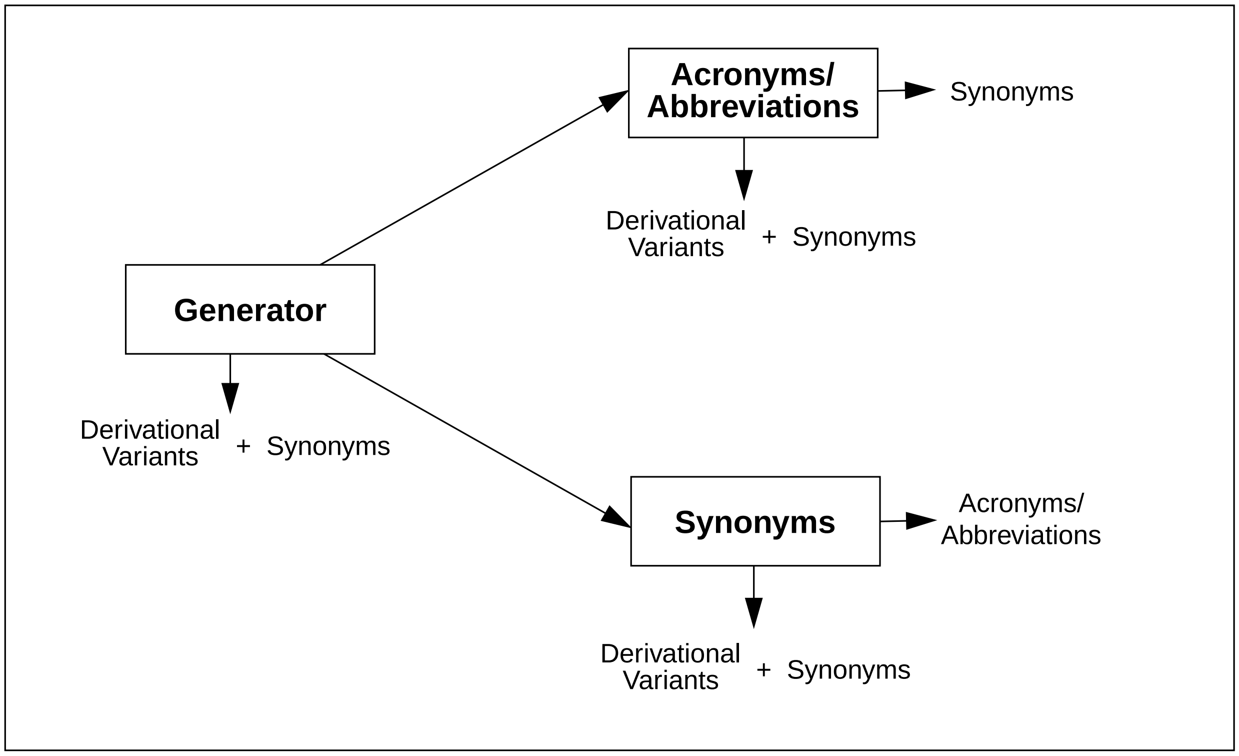
Acronyms/ Abbreviations
Synonyms
Derivational Variants + Synonyms
Generator
Derivational Variants + Synonyms
Synonyms
Acronyms/ Abbreviations
Derivational Variants + Synonyms

Figure 1. Variant Generation

computation for each generator proceeds as follows:

1. Compute all acronyms, abbreviations and synonyms of the generator. This results in the three sets Generator, Acronyms/Abbreviations, and Synonyms which are highlighted with boxes in Figure 1;
2. Augment the elements of the three sets by computing their derivational variants and the synonyms of the derivational variants;
3. For each member of the Acronyms/Abbreviations set, compute synonyms; and
4. For each member of the Synonyms set, compute acronyms/abbreviations.

The issue of whether to recursively generate variants of a given type is handled as follows:

- Acronyms and abbreviations are not recursively generated since doing so almost always produces incorrect results. For example, the abbreviation *na* of *sodium* has expansions *nurse's aide* and *nuclear antigen* which are unrelated to *sodium*; and
- Derivational variants and synonyms are recursively generated since this often produces meaningful variants.

The variants computed for the generator *ocular* are shown in Figure 2. Following each

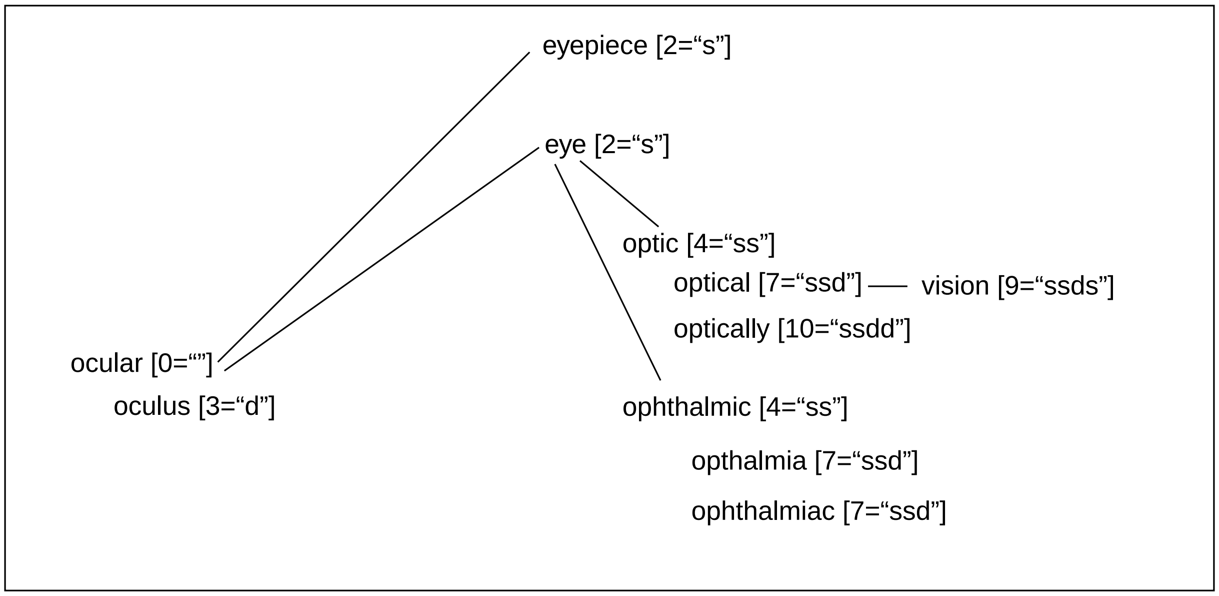

Figure 2. Variants for the generator *ocular*

variant is its variant distance score, a rough measure of how much it varies from its generator (see Section 5) and the history of how it was computed. For example,

- *oculus* (with variant distance 3 and history "d") is simply a derivational variant of the generator *ocular*;
- *optical* (with variant distance 7 and history "ssd") is a derivational variant of a synonym (*optic*) of a synonym (*eye*) of *ocular*; and
- *vision* (with variant distance 9 and history "ssds") is a synonym of the derivational variant *optical* described above.

The variant generation algorithm described here is knowledge intensive. It uses the following knowledge sources:

- the SPECIALIST lexicon and a table of canonical forms derived from it;
- a SPECIALIST knowledge base of acronyms and abbreviations;
- a SPECIALIST knowledge base containing rules of derivational morphology; and
- two knowledge bases of synonyms: one obtained by extracting synonyms from Dorland's Illustrated Medical Dictionary, and a supplemental synonym knowledge base developed for use with SPECIALIST.

4. Meta Candidates

The Meta candidates for a noun phrase consist of the set of all Meta strings containing at least one of the variants computed for the phrase.[1] The candidates are easily found by using a version of the Meta word index, an index from words to all Meta strings containing them. The Meta candidates for the noun phrase *ocular complications* are shown in Figure 3. When a string is not,

```
861 Complications (Complication)
861 complications <1>
638 Eye
611 Optic (Optics)
588 Ophthalmia (Endophthalmitis)
579 Vision
```

Figure 3. Meta Candidates for *ocular complications*

itself, the preferred name for a Meta concept, the preferred name appears in parentheses following the string. The candidates are ordered according to the evaluation function described in the next section. The best candidates are Complications and complications <1> both of which are simple matches involving the head of the phrase.[2] The remaining candidates are variants of *ocular* and are listed in order of similarity to *ocular.*

5. The Evaluation function

The evaluation function computes a measure of the quality of the match between a phrase and a Meta candidate. For normal MetaMap operation the evaluation function is based on four components: *centrality*, *variation*, *coverage*, and *cohesiveness*. A normalized value between 0 (the weakest match) and 1 (the strongest match) is computed for each of these components. A weighted average is computed in which the coverage and cohesiveness components receive twice the weight as the centrality and variation components.[3] The result is normalized to a value between 0 and 1000, 0 indicating no match at all and 1000 indicating an identical match (except for capitalization). When MetaMap is used for browsing (e.g., for term processing), the coverage

1. Here we are assuming normal MetaMap processing where correctness rather than breadth is most important. Since overmatches are rarely good matches, they are ignored for the rest of this discussion. An example of such an overmatch is Postoperative Complications for *ocular complications*. Similarly, Meta strings with gaps (e.g., Computerized Medical Record System for *computerized system*) are also ignored.

2. The symbol <1> denotes the first sense of an ambiguous Meta string. It is ignored during matching.

3. The actual weights used were determined empirically. Also, relative evaluation values were not particularly sensitive to small differences in the weights.

and cohesiveness components are both replaced by a single component, *involvement*. Each of the evaluation function components is discussed below.

- Centrality: The centrality value is simply 1 if the string involves the head of the phrase and 0 otherwise. For the noun phrase *ocular complications*, Complications has centrality value 1; and Eye has value 0.
- Variation: The variation value estimates how much the variants in the Meta string differ from the corresponding words in the phrase. It is computed by first determining the *variation distance* for each variant in the Meta string. This distance is the sum of the distance values for each step taken during variant generation. The values for each step are listed in Table 2. The

Variant Type	**Distance Value**
spelling	0
inflectional	1
synonym or acronym/abbreviation	2
derivational	3

Table 2. Variant Distances

  variation distance determines the variation value for the given variant according to the formula V=4/(D+4). As the total distance value, D, increases from its minimum value of 0, V decreases from a maximum value of 1 and is bounded below by 0. The final variation value for the candidate is the average of the values for each of the variants. For *ocular complications*, Eye has a variant distance value of 2 and hence a variation value of 2/3 (4/(2+4)). Complications has a variant distance value of 0 and hence a variation value of 1.
- Coverage: The coverage value indicates how much of the Meta string and the phrase are involved in the match. In order to compute the value, the number of words participating in the match is computed for both the Meta string and the phrase. These numbers are called the *Meta span* and *phrase span*, respectively. Note, however, that gaps are ignored.[1] The coverage value for the Meta string is the Meta span divided by the length of the string. Similarly, the coverage value for the phrase is the phrase span divided by the length of the phrase. The final coverage value is the weighted average of the values for the Meta string and the phrase where the Meta string is given twice the weight as the phrase. For *ocular complications* and either Eye or Complications, the Meta span and phrase span are both 1, and the coverage value is 5/6 (2/3*(1/1) + 1/3*(1/2)).

1. This somewhat surprising scheme is illustrated by the following example. In computing the coverage for the phrase “an inferior vena caval stent filter” with the Meta string Inferior Vena Cava Filter, the phrase span is 5 even though “stent” does not participate in the match.

- Cohesiveness: The cohesiveness value is similar to the coverage value but emphasizes the importance of connected components. A connected component is a maximal sequence of contiguous words participating in the match. The connected components for both the Meta string and the phrase are computed. This information is abstracted by noting the size of each component. This produces a set of connected component sizes for both the Meta string and the phrase. The cohesiveness value for the Meta string is the sum of the squares of the connected Meta string component sizes divided by the square of the length of the string. A similar cohesiveness value is computed for the phrase. The final cohesiveness value is the weighted average of the Meta string and phrase values where the Meta string is again given twice the weight as the phrase. For *ocular complications* and either Eye or Complications, the connected component sizes for both the Meta string and the phrase are {1} since one word from either the phrase or Meta string participates in the match. The cohesiveness value is 3/4 (2/3*(1/1) + 1/3(1/4)).

The final evaluation for Eye is the weighted average (0 + 2/3 + 2*(5/6) + 2*(3/4))/6 which normalizes to 638. Similarly, the final evaluation for Complications is (1 + 1 + 2*(5/6) + 2*(3/4))/6 which normalizes to 861.

- Involvement: The involvement value is a rough approximation of the coverage and cohesiveness values. The strict word order implied by the matchmap is no longer followed. The involvement value for the phrase is the proportion of phrase words which *can* map to a Meta word whether or not they do according to the matchmap. For example, given the phrase *Advanced cancer of the lung* with words [advanced, cancer, lung] and the Meta string “Lung Cancer” with words [lung, cancer], the matchmap maps lung to lung, but does not map cancer because of word order. The phrase involvement value here is 2/3 as opposed to the coverage value of 1/3. Similarly, the involvement value for the Meta string is the proportion of words which *can* be mapped to from the phrase. For the current example, the Meta involvement value is 2/2 or 1 rather than 1/2 for coverage. Thus the final involvement value for this example is the weighted average (2/3 + 1)/2 or 0.83.[1]

6. The Final Mapping

The final step in the mapping algorithm is straightforward. It consists of examining combinations of Meta candidates which participate in matches with disjoint parts of the noun phrase. The evaluation function is applied to the combined candidates, and the best ones form the final mapping result. The best mappings for *ocular complications* are shown in Figure 4. The centrality, variation, coverage and cohesiveness values for the mapping are 1, 2/3, 1 and 1, respectively. The final evaluation of the mapping is the weighted average (1 + 2/3 + 2*1 + 2*1)/6 which normalizes to 861[2] and is reported as a confidence value in the figure.

1. Note that the weighting of phrase involvement and Meta involvement is equal rather than the normal 1:2 ratio.
2. It is coincidence that this is the same value as the candidate score for Complications.

```
[mod([tokens([ocular]), metaconc([Eye])]),
 head([tokens([complications]), metaconc([Complications])]),
 confid(861)]
   and
[mod([tokens([ocular]), metaconc([Eye]),
 head([tokens([complications]), metaconc([complications <1>]),
 confid(861))]
```

Figure 4. The Best Meta Mappings for *ocular complications*

7. MetaMap Control Options

MetaMap behavior is controlled by several option flags each of which has a short version (e.g., `-p`) and a long version (e.g., `--plain_syntax`). With the exception of the `--threshold` option, each option is a toggle switch. Specifying a default option toggles it off; specifying a non-default option toggles it on. The options are described in the following sections.

7.1 The default options

MetaMap's default behavior is defined by the options: **`-t`** (**`--tag_text`**), **`-l`** (**`--stop_large_n`**), **`-b`** (**`--best_mappings_only`**), **`-p`** (**`--plain_syntax`**), **`-c`** (**`--candidates`**), **`-s`** (**`--semantic_types`**), and **`-m`** (**`--mappings`**). Each of these options is defined below.

7.2 Processing options

Processing options control MetaMap's internal behavior.

- **`-t`** (**`--tag_text`**) specifies that the SPECIALIST parser will use the results of the Xerox Parc Part-of-Speech Tagger to assist in parsing. If a preprocessed tag file is specified on the command line, tagging results are read from it; otherwise, tagging is done dynamically using the server version of the tagger.
- `-z` (`--term_processing`) invokes MetaMap's browsing mode and causes it to use the involvement metric rather than coverage and cohesiveness for evaluating Meta candidates. It is normally used in conjunction with the `--allow_overmatches` and `--allow_concept_gaps` options.
- `-o` (`--allow_overmatches`) causes MetaMap to retrieve Meta candidates which are overmatches. This greatly increases the number of candidates retrieved and is consequently much

slower than MetaMap without overmatches. The `--allow_overmatches` option is appropriate for browsing purposes.

- `-g` (`--allow_concept_gaps`) causes MetaMap to retrieve Meta candidates with gaps (such as Unspecified childhood psychosis for *unspecified psychosis*). The `--allow_concept_gaps` option is appropriate for browsing purposes.
- `-a` (`--no_acros_abbrs`) prevents the generation of acronym/abbreviation variants.
- `-u` (`--unique_acros_abbrs_only`) restricts the generation of acronym/abbreviation variants to those acronyms and abbreviations with unique expansions.
- **`-l`** (**`--stop_large_n`**) prevents retrieval of Meta candidates based on either a two-character word occurring in more than 1,000 Meta strings or a one-character word occurring in more than 500 Meta strings.

7.3 Output options

- **`-b`** (**`--best_mappings_only`**) restricts mappings displayed to only the top scoring ones.
- `-r` (`--threshold <integer>`) restricts output to candidates with evaluation score of the threshold or better.
- `-j` (`--mesh_projection`) is a special option for restricting MetaMap to the MeSH vocabulary.
- `-q` (`--machine_output`) causes output to take the form of Prolog clauses rather than human-readable form. The `--machine_output` option affects all other output options.
- **`-p`** (**`--plain_syntax`**) and `-x` (`--syntax`) control the output form of the results of the SPECIALIST minimal commitment parser. **`--plain_syntax`** simply outputs text; `--syntax` outputs a Prolog-like structure showing details of the syntactic processing.
- `-v` (`--variants`) and `-f` (`--full_variants`) display summary (`--variants`) and detailed (`--full_variants`) information regarding variant generation.
- **`-c`** (**`--candidates`**) causes the list of Meta candidates to be displayed.
- `-n` (`--number_the_candidates`) simply numbers the displayed candidates.
- **`-s`** (**`--semantic_types`**) causes the semantic types of Meta concepts to be displayed.
- **`-m`** (**`--mappings`**) causes mappings to be displayed.

7.4 Miscellaneous options

- `-h` (`--help`) displays MetaMap usage.
- `-i` (`--info`) causes system information to be displayed.
- `-w` (`--warnings`) enables the display of conditions which are noteworthy if not erroneous.